

本贴由 ouravr 网友 fsmcu 发布,本人拜读后整理上传,感谢 fsmcu 的无私奉献,原帖连接:

[http://www.ouravr.com/bbs/bbs\\_content\\_all.jsp?bbs\\_sn=936512](http://www.ouravr.com/bbs/bbs_content_all.jsp?bbs_sn=936512)

Blackcafe @ Apr 30,2008

对于温度控制系统参数的整定过程及方法, 搜索了很多论坛, 大家都只提到 PID 数学原型, 很少看到有实验过程及  $K_p, K_i, K_d$  参数的整定过程及方法, 这 2 天闲着没事, 来做个温度控制实验, 并将一些实验过程写上来, 我也是正在摸索阶段, 希望大家踊跃发言哟, 各位有好的建议尽管提出来, 然后我来实验, 将实验数据整理上来, 希望对于以后想做温度控制的朋友有所帮助

硬件:

1. 用可控硅控制 200W 加热丝, 对一铁块(重量 2 斤)加热, 用 K 型热电偶采集温度, 采用 MAX6675 做温度转换, 可以到 0.25 度的精度, 并且外围很简单, 很容易与 CPU 对接, 采用 SPI 通信, 读取当前的温度值
2. 过零检测电路, 将交流信号全波整流后得到的波形去控制 NPN 管, 将信号整形后接到 CPU 外部中断脚, 为系统提供基准时间, 所以 CPU 中断的频率是 100HZ

实验目标:

在 100 度到 200 度内可对任意设定的温度恒温, 精度先做到  $\pm 1$  度吧

基本的控制实现方法:

因为是对加热的铁块温度进行控制, 属于滞后效应系统, 所以采样周期先定为 5 秒(这里指的是 PID 计算的周期, 注意我的温度采样是时时的), 所以 CPU 外部中断次数为 500 次, 对应的 PID 计算结果输出为 0~500, 就是说把这 5 秒钟划分为 500 等份, 根据计算的结果来决定在这 5 秒钟内应该加热多少等份

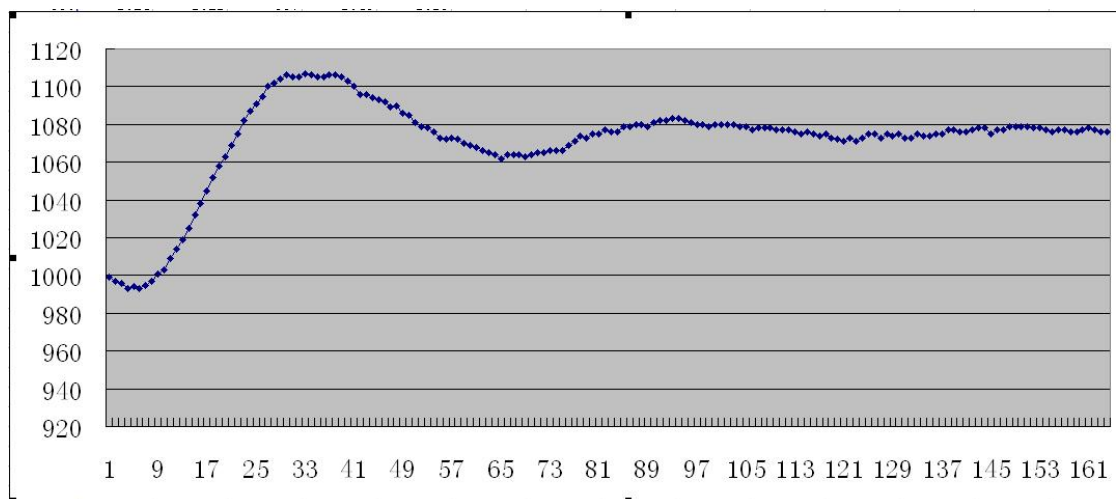
软件:

采用 PID 控制方法, 我先采用位置式输出方式, 公式原型:  $u(t) = k_p * e(t) + k_i * [e(1) + e(2) + \dots + e(t)] + k_d * [e(t) - e(t-1)]$ , 这里先做基本的 PID 算法, 达到控制目标后再来优化算法提高恒温精度, 考虑到实验温度过高实验时间会过长, 所以我先定目标控制温度为 110 度, 等控制好了再看其他温度会达到多少精度, 为了提高加热速度提前 20 度开始 PID 控温

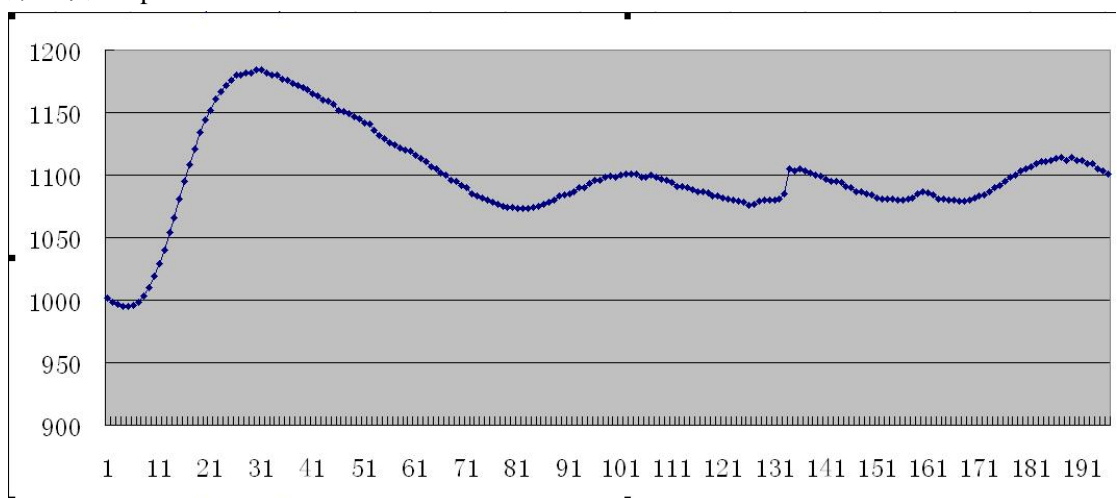
下面是调节参数的过程及数据:

参照网上一些方法, 先确定  $K_p$ , 即令  $K_i, K_d=0$ , 只用比例调节, 得到一个稳定的越接近控制目标的震荡参数, 然后根据这个  $K_p$  和震荡周期来计算  $T_i, T_d$ ,

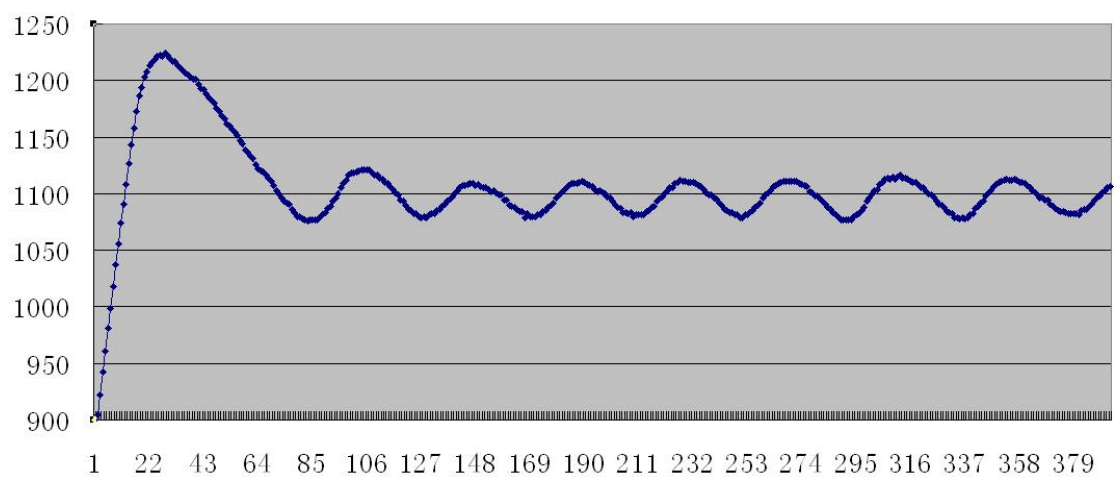
第 1 次:  $K_p=2.5$ , 测试数据如下见图片: Y 坐标为温度值, 放大了 10 倍, X 坐标为时间每 5 秒一个点,



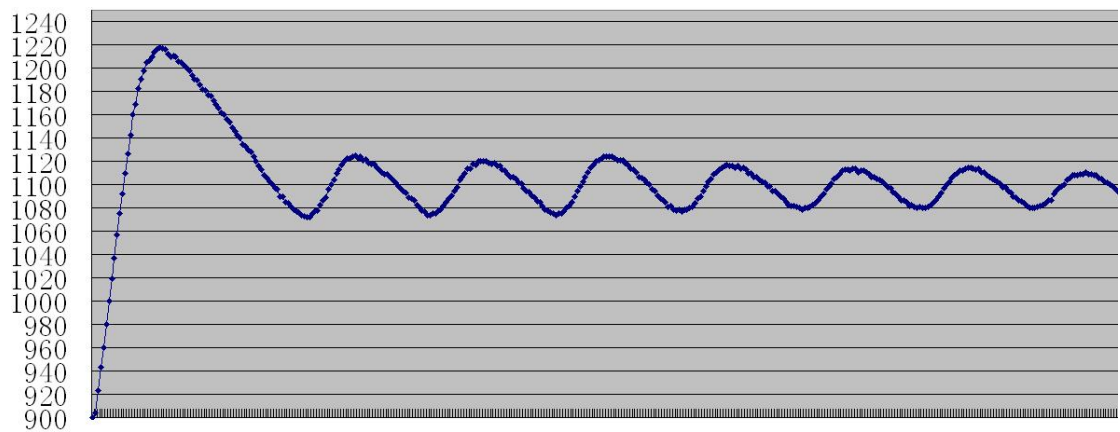
第 2 次:  $K_p=5$



第 3 次:  $K_p=8.5$ , 这次的测试时间比较长, 因为比较接近稳定震荡了, 图片如下:

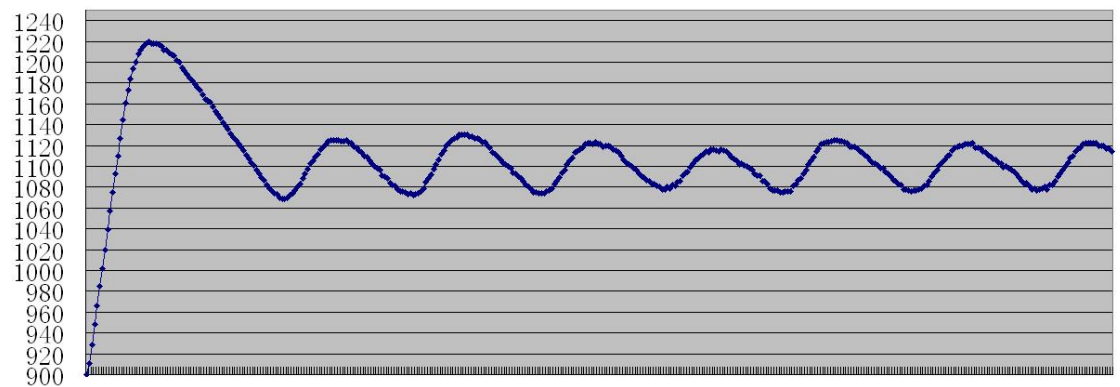


第 4 次:  $K_p=9$ , 从图片已经基本稳定, 下次再测试一下 9.3 这个值



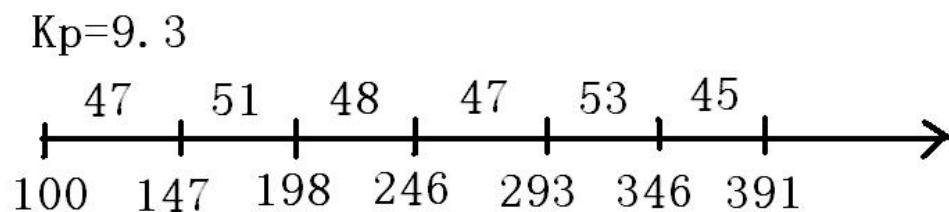
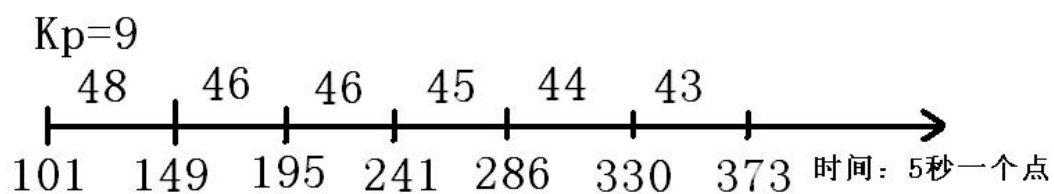
1 21 41 61 81 101 121 141 161 181 201 221 241 261 281 301 321 341 361 38

第 5 次实验结果,  $K_p=9.3$ , 从图上分析, 震荡幅度稍比 9 的大点, 但是上下幅度比较均匀, 所以先确定  $K_p$  为 9~9.3 之间



1 20 39 58 77 96 115 134 153 172 191 210 229 248 267 286 305 324 343 362 381 40

现在整理了一下  $K_p=9$  和  $K_p=9.3$  的震荡周期, 图片分析如下



从数据看震荡周期基本都是 48~43 个点, 每个点是 5 秒, 所以震荡周期为 225 秒

接下来根据下面典型参数计算表:

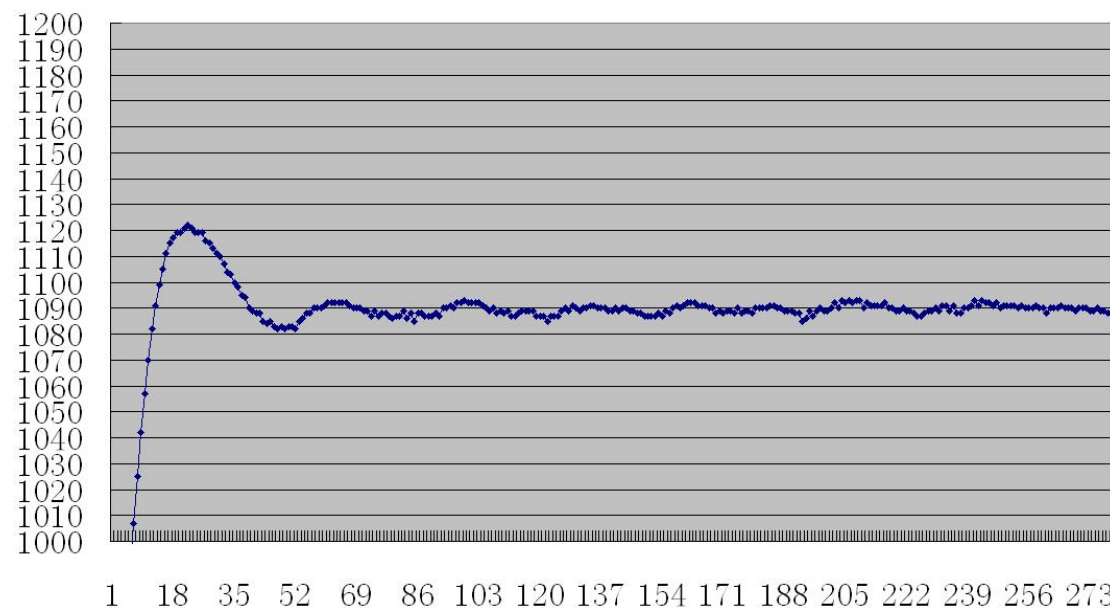
### Ziegler-Nichols 参数

控制器	$K_p$	$T_i$	$T_d$
P	$0.50 \cdot K_c$	/	/
PD	$0.65 \cdot K_c$	/	$0.12 \cdot P_c$
PI	$0.45 \cdot K_c$	$0.85 \cdot P_c$	/
PID	$0.65 \cdot K_c$	$0.5 \cdot P_c$	$0.12 \cdot P_c$

我先做 PD 实验：

根据临界增益  $K_C=9.2$  振荡周期  $P_c=225$  秒，计算出  $K_p=5.98$ ， $T_d=27$  秒，公式  $K_d=K_p \cdot T_d/T=5.98 \cdot 27/5=32.392$

实验结果出来了，见下图



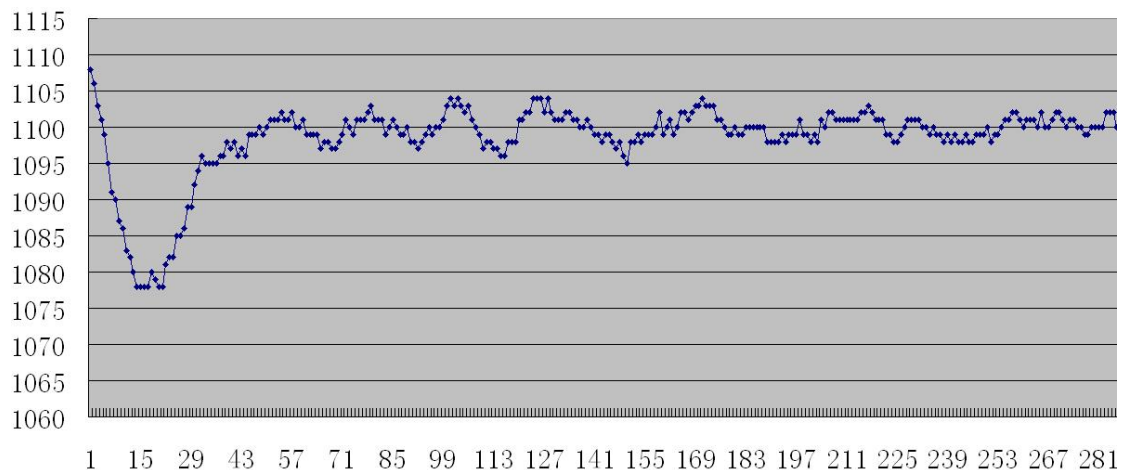
从图片数据分析最终稳定温度为：109 度，上下波动为 0.5 度，而我的设定目标温度为 110，说明这里引入了 2 度的稳态误差  
需要加入积分调节器才能消除这个稳态误差

计算  $T_i=112.5$  秒，根据公式： $K_i=K_p \cdot T/T_i=5.98 \cdot 5/112.5=0.265$

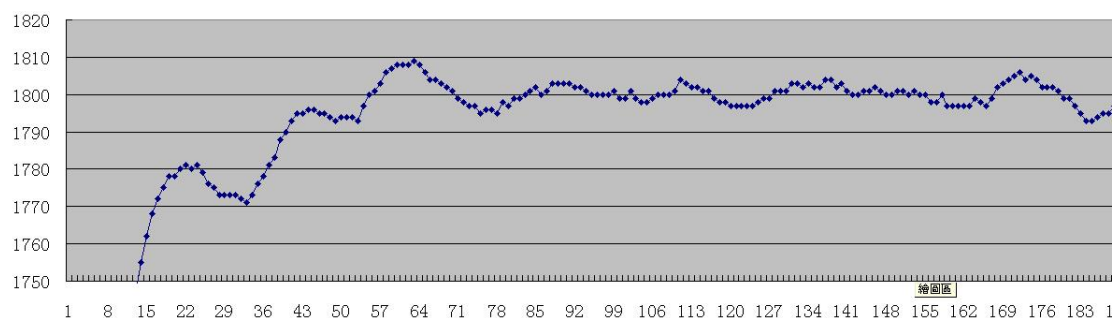
下面加入 PID 调节，看实验结果如何？

下面是加入积分分离法的 PID 算法实验结果

从图中数据看出已经符合设计要求，在 110 度稳定精度为  $\pm 0.5$ ，等一下看看其他设定温度的控制效果

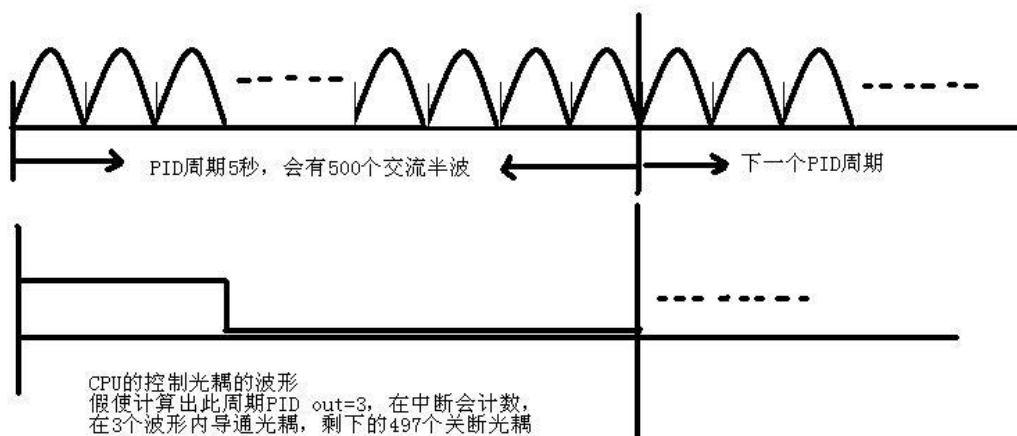


下面是 180 度目标的控温效果，恒温时为 180， $\pm 0.5$  度的误差，基本已经满足要求



以上表格生成方法：将采集的时时温度传到 PC，然后导入到 EXCEL 中再用图表分析功能做成的

我的硬件系统是 51 单片机通过光耦触发可控硅来实现给电热丝供电，并且检测交流电过零，控制导通周期，至于上面看到的数据解析图片，是单片机通过串口将时时温度数据传递给 PC，然后保存起来用 EXCEL 软件分析，温度走势，看 PID 的控温效果，等一下我将实验图片上传。下面是控制解析图，不知道表达清楚不？



PID 的源代码在网上基本都是公开的，既然大家都希望看一看，我还是贴出来吧，估计看了也很失望，因为我的也是网上的代码一模一样的，既然数学模型已经建立，程序只是把这个模型用语言把他表达出来而已，其实很多人 PID 实验不成功，就是没有仔细去分析参数的调节过程，我是想通过上面的实验过程给大家一个思路去如何调节这几个参数，而不是在网上到处搜索源程序，然后抱怨下载的 PID 源程序都不能满足自己的调节要求。

经过实验发现：位置式整定的参数不能直接用在增量式上面，会出现超调的时间居多，无法稳定在目标值上

```
//PID 计算
UINT16 PID_Cal(void)
{
    float xdata pterm,iterm,dterm;

    //把当前的温度和设定的目标温度定义好
    ppid->pv = SystemRealTemprature;
    ppid->sp = SystemSetTemprature;

    //根据实验得出:Kp=5.98 T=5 Td=27 Ti=112.5

    //PID 数学模型
    //u(t) = kp * e(t) + ki * [e(1) + e(2) + ....+ e(t)] + kd * [e(t) - e(t-1)]

    ppid->errk = (float)(ppid->sp - ppid->pv);

    pterm = ppid->kp * ppid->errk;

    ppid->sum += ppid->errk;
    iterm = ppid->ki * ppid->sum;

    dterm = ppid->kd * (ppid->errk - ppid->errk_1);

    ppid->Last_Out = pterm + iterm + dterm;

    ppid->errk_2 = ppid->errk_1;
    ppid->errk_1 = ppid->errk;

    if(ppid->Last_Out > SampleT)
        ppid->Last_Out = SampleT;
    else if(ppid->Last_Out < 0)
        ppid->Last_Out = 0;

    printf("%d    p=%d    i=%d    d=%d    ",SystemRealTemprature, (int)pterm,
(int)iterm, (int)dterm);
    printf("H=%d\n", (UINT16)ppid->Last_Out);

    return      (UINT16)ppid->Last_Out;
}
```